UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/762,621	01/22/2004	Andrew Hadley	03-0023	5052

24319 7590 12/06/2007

LSI CORPORATION
1621 BARBER LANE
MS: D-106
MILPITAS, CA 95035

EXAMINER	
CHU, GABRIEL L	

ART UNIT	PAPER NUMBER
2114	

MAIL DATE	DELIVERY MODE
12/06/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No. 10/762,621	Applicant(s) HADLEY ET AL.	
	Examiner Gabriel L. Chu	Art Unit 2114	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on 04 September 2007.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) 12,14-21 and 44-50 is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) 12,14-21 and 44-50 is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on 22 January 2004 is/are: a)☐ accepted or b)☒ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

Drawings

1. The drawings are objected to under 37 CFR 1.83(a). The drawings must show every feature of the invention specified in the claims. Therefore, the “states”, “layer/level of abstraction" and "testing, validating, and simulating a device under test" must be shown or the feature(s) canceled from the claim(s). No new matter should be entered.

Corrected drawing sheets in compliance with 37 CFR 1.121(d) are required in reply to the Office action to avoid abandonment of the application. Any amended replacement drawing sheet should include all of the figures appearing on the immediate prior version of the sheet, even if only one figure is being amended. The figure or figure number of an amended drawing should not be labeled as “amended.” If a drawing figure is to be canceled, the appropriate figure must be removed from the replacement sheet, and where necessary, the remaining figures must be renumbered and appropriate changes made to the brief description of the several views of the drawings for consistency. Additional replacement sheets may be necessary to show the renumbering of the remaining figures. Each drawing sheet submitted after the filing date of an application must be labeled in the top margin as either “Replacement Sheet” or “New Sheet” pursuant to 37 CFR 1.121(d). If the changes are not accepted by the examiner, the applicant will be notified and informed of any required corrective action in the next Office action. The objection to the drawings will not be held in abeyance.

Claim Objections

2. Applicant is reminded of 37 C.F.R. 1.75 (d)(1) which states that the claim or claims must conform to the invention as set forth in the remainder of the specification and the terms and phrases used in the claims must find clear support or antecedent basis in the description so that the meaning of the terms in the claims may be ascertainable by reference to the description. (See 1.58(a).). The specification refers to a "level" of abstraction, but a "layer" is not found.

3. Claim 50 is objected to because of the following informalities: Referring to claim 50, "status machine" is understood to refer to "state machine". Appropriate correction is required.

Claim Rejections - 35 USC § 112

4. The following is a quotation of the first paragraph of 35 U.S.C. 112:

> The specification shall contain a written description of the invention, and of the manner and process of making and using it, in such full, clear, concise, and exact terms as to enable any person skilled in the art to which it pertains, or with which it is most nearly connected, to make and use the same and shall set forth the best mode contemplated by the inventor of carrying out his invention.

5. **Claims 12, 14-21, 44-50 rejected under 35 U.S.C. 112, first paragraph, as failing to comply with the enablement requirement.** The claim(s) contains subject matter which was not described in the specification in such a way as to enable one skilled in the art to which it pertains, or with which it is most nearly connected, to make and/or use the invention. Referring to claim 12, and subsequently claims 14-21, 44-50, the specification is not sufficiently descriptive of "applying a layer of abstraction to multiple parameters". Further, reading the specification, Examiner is unable to ascertain what in the specification specifically corresponds to any such abstraction other than a

brief mention in the second paragraph of the summary. One of ordinary skill in the art would not be able to make and use this invention from this description.

Claim Rejections - 35 USC § 101

6. 35 U.S.C. 101 reads as follows:

> Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

Claims 12, 14-21, 44-50 rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter. Referring to claim 12, and subsequently claims 14-21, 44-50, Applicant claims instructions "storable" on a medium executed by a computer. Storable instructions are not necessarily stored. Furthermore, this is understood to be a claim to instructions per se.

Claim Rejections - 35 USC § 102

7. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

8. Claims 12, 21, 44, 45, 47-49 rejected under 35 U.S.C. 102(e) as being anticipated by US 20040078674 to Raimi et al.

9. Referring to claim 12, Raimi discloses

a state machine which is "capable of executing at least one function" (Figures 1-

3.),

the function being "editing, storing, loading, [or] displaying" (Paragraph 60, "fetch", paragraph 145, "load", paragraph 149, "read/write".),

the function having "code common" to multiple parameters, the multiple parameters having had applied thereto "a layer of abstraction" (Abstract, "Designers can specify architectural states using Boolean variables and generate test vectors for any state that is reachable, within the compute resources available or prove that the state is unreachable within the given bound k. This technique also allows the designer to restrict specific states from being covered by the test. In a general way, this technique allows the user to describe constraints an instruction sequence should obey and then to generate sequences which obey them." Paragraph 58, "FIG. 6 outlines a process 600 of creating a test case once a state transition system modeling the ISA has been defined. Once such a state transition system, along with assumed constraints has been created, one can specify the types of state sequences one would like to determine are possible as formulae in a temporal logic as in step 638, and use SAT-based bounded model checking techniques, as in step 640, to find examples of such state sequences, or, alternatively, prove that these are impossible. For any state sequence that is returned by the Boolean satisfiability-solving tool, such a sequence may be considered to be an instruction sequence template. It is a template and not a complete instruction sequence because in the general case, not all aspects of the instruction opcodes will be specified, because not all of these will have been modeled with the Boolean variables used for representing instructions and relationships among instructions. Rather, there will be a

degree of freedom in choosing among various instructions and among various parameters for these instructions, such that any of these choices will result in the same architectural states being reached. Next, in step 642, the instruction sequence template and the constraints on relationships among instructions are transferred into a specialized test generation program, a test program generator specific to the given ISA that generates assembly language test programs based on user specified sequences of opcode types and user specified constraints. Such a test program generator then, in step 644, instantiates fully specified opcodes into the instruction sequence template, in order to create an executable, assembly language test program that, when executed on a real hardware or software implementation of the ISA will reach the architectural states of interest."),

for "testing, validating, [or] simulating a device under test" (Abstract, "verification".).

10. Referring to claim 21, Raimi discloses at least one of the multiple parameters is "independent of type" (Paragraph 58, "Rather, there will be a degree of freedom in choosing among various instructions and among various parameters for these instructions, such that any of these choices will result in the same architectural states being reached." Wherein the parameters are not disclosed to be dependent on a bus.).

11. Referring to claim 44, Raimi discloses adding at least one new parameter to the multiple parameters; and applying the same layer of abstraction to the at least one new parameter that was applied to the multiple parameters (Abstract, "Designers can specify architectural states using Boolean variables and generate test vectors for any state that

is reachable, within the compute resources available or prove that the state is unreachable within the given bound k. This technique also allows the designer to restrict specific states from being covered by the test. In a general way, this technique allows the user to describe constraints an instruction sequence should obey and then to generate sequences which obey them." Paragraph 58, "FIG. 6 outlines a process 600 of creating a test case once a state transition system modeling the ISA has been defined. Once such a state transition system, along with assumed constraints has been created, one can specify the types of state sequences one would like to determine are possible as formulae in a temporal logic as in step 638, and use SAT-based bounded model checking techniques, as in step 640, to find examples of such state sequences, or, alternatively, prove that these are impossible. For any state sequence that is returned by the Boolean satisfiability-solving tool, such a sequence may be considered to be an instruction sequence template. It is a template and not a complete instruction sequence because in the general case, not all aspects of the instruction opcodes will be specified, because not all of these will have been modeled with the Boolean variables used for representing instructions and relationships among instructions. Rather, there will be a degree of freedom in choosing among various instructions and among various parameters for these instructions, such that any of these choices will result in the same architectural states being reached. Next, in step 642, the instruction sequence template and the constraints on relationships among instructions are transferred into a specialized test generation program, a test program generator specific to the given ISA that generates assembly language test programs based on user specified sequences of

opcode types and user specified constraints. Such a test program generator then, in step 644, instantiates fully specified opcodes into the instruction sequence template, in order to create an executable, assembly language test program that, when executed on a real hardware or software implementation of the ISA will reach the architectural states of interest.").

12. Referring to claim 45, Raimi discloses performing one selected from adding, changing, and deleting at least one state from the state machine (Figure 3, 310.).

13. Referring to claim 47, Raimi discloses building the state machine further comprises importing parameter information into the code common to the multiple parameters and specific to the at least one function (Figure 3, 310-314.).

14. Referring to claim 48, Raimi discloses the state machine is independent of bus type (Raimi does not disclose the state machine is dependent on any particular bus.).

15. Referring to claim 49, Raimi discloses a second state machine interrelated with the state machine (A state machine, particularly of the type disclosed in Raimi, is comprised of state machines.).

Claim Rejections - 35 USC § 103

16. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

17. **Claim 14 rejected under 35 U.S.C. 103(a) as being unpatentable over US 20040078674 to Raimi et al. as applied to claim 12 above, and further in view of US 20030093608 to Jaramillo et al.**

18. Referring to claim 14, although Raimi does not specifically disclose the multiple parameters include a PCI cache line size, adjusting a PCI cache line size is known in the art. An example of this is shown by Jaramillo from paragraph 44, "This approach can be implemented by using other multiples or with a programmable multiple, or the standard PCI specification cache line size register can be adjusted such that the PCI to PCI bridge 350 actually prefetches multiple cache lines." A person of ordinary skill in the art at the time of the invention would have been motivated to use PCI cache line size as a parameter because, from paragraph 44 of Jaramillo, "It raises the overall system performance dramatically." Further, such a parameter would have been included for testing because Raimi discloses from the field of invention that the invention is broadly applicable to the testing of digital hardware, and further, from paragraph 145, "Since it is often not necessary to describe data values completely to describe the effects of instruction execution, it is often necessary to represent a few bits among the input or target values for an instruction, as needed, and to represent architectural features, such as an instruction's target destination, the execution unit chosen, or value of conditional flags produced. FIG. 9 shows an exemplary output of the process 800, describing a sequence of instruction types 900 obeying certain constraints including a load (LD) 910, subtract (Sub) 920, and addition (Add) 930. A test generator based on the underlying instruction set architecture, the BOPS 2040 shown in FIG. 1 and FIG. 2, is then used to

generate an assembly language test program that has the same instruction types in sequence, and satisfies the given constrains, such as certain opcode choices, ability to set certain bits in a target register, and the like."

19. **Claim 15 rejected under 35 U.S.C. 103(a) as being unpatentable over US 20040078674 to Raimi et al. as applied to claim 12 above, and further in view of US 6675244 to Elliot et al.**

20. Referring to claim 15, although Raimi does not specifically disclose the multiple parameters include a Small Computer System Interface (SCSI) synchronous rate, adjusting the SCSI system rate is known in the art. An example of this is shown by Elliot from line 34 of column 7, "The ASRT state 608 is a transitory state in which the timer is loaded with a value suitable for a delay discussed in conjunction with the next state, a WAIT_ASRT state 610. The value loaded into the timer during the ASRT state 608 depends on whether the linear mode is enabled, what the determined SCSI synchronous rate is, and whether this particular clock pulse is being "stretched". These aspects are further discussed below in conjunction with FIGS. 8-12. To summarize, if the linear mode is enabled, the SCSI clock will be asserted for a number of repeater 40 clock cycles that most closely matches the incoming clock signal from the other side of the repeater 40, but with some degree of "snapping" when the rate is near a standard SCSI rate." A person of ordinary skill in the art at the time of the invention would have been motivated to include a SCSI synchronous rate because, as disclosed by Elliot, the rate affects system performance. Further, such a parameter would have been included for testing because Raimi discloses from the field of invention that the invention is

broadly applicable to the testing of digital hardware, and further, from paragraph 145, "Since it is often not necessary to describe data values completely to describe the effects of instruction execution, it is often necessary to represent a few bits among the input or target values for an instruction, as needed, and to represent architectural features, such as an instruction's target destination, the execution unit chosen, or value of conditional flags produced. FIG. 9 shows an exemplary output of the process 800, describing a sequence of instruction types 900 obeying certain constraints including a load (LD) 910, subtract (Sub) 920, and addition (Add) 930. A test generator based on the underlying instruction set architecture, the BOPS 2040 shown in FIG. 1 and FIG. 2, is then used to generate an assembly language test program that has the same instruction types in sequence, and satisfies the given constrains, such as certain opcode choices, ability to set certain bits in a target register, and the like."

21. **Claim 16 rejected under 35 U.S.C. 103(a) as being unpatentable over US 20040078674 to Raimi et al. as applied to claim 12 above, and further in view of "block size" by Microsoft Computer Dictionary (MSCD).**

22. Referring to claim 16, although Raimi does not specifically disclose the multiple parameters include block size, adjusting the block size is known in the art. An example of this is shown by MSCD, "The declared size of a block of data transferred internally within a computer, via FTP, or by modem. The size is usually chosen to make most efficient use of all the hardware devices involved." A person of ordinary skill in the art at the time of the invention would have been motivated to include a block size because, as disclosed by MSCD, the block size affects system performance. Further, such a

parameter would have been included for testing because Raimi discloses from the field of invention that the invention is broadly applicable to the testing of digital hardware, and further, from paragraph 145, "Since it is often not necessary to describe data values completely to describe the effects of instruction execution, it is often necessary to represent a few bits among the input or target values for an instruction, as needed, and to represent architectural features, such as an instruction's target destination, the execution unit chosen, or value of conditional flags produced. FIG. 9 shows an exemplary output of the process 800, describing a sequence of instruction types 900 obeying certain constraints including a load (LD) 910, subtract (Sub) 920, and addition (Add) 930. A test generator based on the underlying instruction set architecture, the BOPS 2040 shown in FIG. 1 and FIG. 2, is then used to generate an assembly language test program that has the same instruction types in sequence, and satisfies the given constrains, such as certain opcode choices, ability to set certain bits in a target register, and the like."

23. Claim 17-19 rejected under 35 U.S.C. 103(a) as being unpatentable over US 20040078674 to Raimi et al. as applied to claim12 above, and further in view of US 6507842 to Grey et al.

24. Referring to claim 17, Raimi discloses values for the multiple parameters (from paragraph 145, "Since it is often not necessary to describe data values completely to describe the effects of instruction execution, it is often necessary to represent a few bits among the input or target values for an instruction, as needed, and to represent architectural features, such as an instruction's target destination, the execution unit

chosen, or value of conditional flags produced. FIG. 9 shows an exemplary output of the process 800, describing a sequence of instruction types 900 obeying certain constraints including a load (LD) 910, subtract (Sub) 920, and addition (Add) 930. A test generator based on the underlying instruction set architecture, the BOPS 2040 shown in FIG. 1 and FIG. 2, is then used to generate an assembly language test program that has the same instruction types in sequence, and satisfies the given constrains, such as certain opcode choices, ability to set certain bits in a target register, and the like.").

Although Raimi does not specifically disclose that those values may be drawn from a "look-up table of default values", such a table is known in the art. An example of this is shown in Grey, from line 3 of column 3, "After including the Property Loader step in a sequence at edit time, the user may configure the step to load the desired variable and/or property values from the database. In various embodiments, this configuration process may require specifying various types of information. For example, the user may specify a particular database from which to load the values. The user may also specify a mapping of properties/variables to database values. For example, specifying this mapping may comprise specifying a database table and a mapping of properties/variables to columns in the table. For other types of databases, e.g., object-oriented databases, this mapping may be specified in any of various other ways. Exemplary user interface dialog boxes for specifying the mapping are discussed. The user may also configure the Property Loader step with filtering information specifying criteria which the database values must satisfy in order to be loaded. If the filtering criteria are not specified, default values for the variables/properties may be used

instead." A person of ordinary skill in the art at the time of the invention could have been motivated to use such a default value database because it provides values for parameters for which criteria are not specified, or at least not fully specified, as is the case in Raimi, as cited above.

25. Referring to claim 18, Raimi discloses values for the multiple parameters (from paragraph 145, "Since it is often not necessary to describe data values completely to describe the effects of instruction execution, it is often necessary to represent a few bits among the input or target values for an instruction, as needed, and to represent architectural features, such as an instruction's target destination, the execution unit chosen, or value of conditional flags produced. FIG. 9 shows an exemplary output of the process 800, describing a sequence of instruction types 900 obeying certain constraints including a load (LD) 910, subtract (Sub) 920, and addition (Add) 930. A test generator based on the underlying instruction set architecture, the BOPS 2040 shown in FIG. 1 and FIG. 2, is then used to generate an assembly language test program that has the same instruction types in sequence, and satisfies the given constrains, such as certain opcode choices, ability to set certain bits in a target register, and the like.").

Although Raimi does not specifically disclose these values may come from a look-up table providing type and value information for each of the multiple parameters, this is known in the art. An example of this is shown in Grey, from line 3 of column 3, "After including the Property Loader step in a sequence at edit time, the user may configure the step to load the desired variable and/or property values from the database. In various embodiments, this configuration process may require specifying

various types of information. For example, the user may specify a particular database from which to load the values. The user may also specify a mapping of properties/variables to database values. For example, specifying this mapping may comprise specifying a database table and a mapping of properties/variables to columns in the table. For other types of databases, e.g., object-oriented databases, this mapping may be specified in any of various other ways. Exemplary user interface dialog boxes for specifying the mapping are discussed. The user may also configure the Property Loader step with filtering information specifying criteria which the database values must satisfy in order to be loaded. If the filtering criteria are not specified, default values for the variables/properties may be used instead." A person of ordinary skill in the art at the time of the invention could have been motivated to use such a value database because it provides values for parameters for which criteria which may be specified or partially specified, as cited by Raimi above.

26. Referring to claim 19, Raimi in view of Grey discloses the type and value information includes a range of values that are permitted for each of the multiple parameters (from paragraph 145 of Raimi (with emphasis), "Since it is often not necessary to describe data values completely to describe the effects of instruction execution, it is often necessary to represent a few bits among the input or target values for an instruction, as needed, and to represent architectural features, such as an instruction's target destination, the execution unit chosen, or value of conditional flags produced. FIG. 9 shows an exemplary output of the process 800, describing a sequence of instruction types 900 obeying certain constraints including a load (LD) 910,

subtract (Sub) 920, and addition (Add) 930. A test generator based on the underlying instruction set architecture, the BOPS 2040 shown in FIG. 1 and FIG. 2, is then used to generate an assembly language test program that has the same instruction types in sequence, and satisfies the given **constrains**, such as certain opcode choices; ability to set certain bits in a target register, and the like." Grey, from line 3 of column 3, with emphasis, "After including the Property Loader step in a sequence at edit time, the user may configure the step to load the desired variable and/or property values from the database. In various embodiments, this configuration process may require specifying various types of information. For example, the user may specify a particular database from which to load the values. The user may also specify a mapping of properties/variables to database values. For example, specifying this mapping may comprise specifying a database table and a mapping of properties/variables to columns in the table. For other types of databases, e.g., object-oriented databases, this mapping may be specified in any of various other ways. Exemplary user interface dialog boxes for specifying the mapping are discussed. The user may also configure the Property Loader step with filtering information specifying **criteria** which the database values must satisfy in order to be loaded. If the filtering criteria are not specified, default values for the variables/properties may be used instead.").

27. **Claim 20 rejected under 35 U.S.C. 103(a) as being unpatentable over US 20040078674 to Raimi et al. and US 6507842 to Grey et al. as applied to claim 19 above, and further in view of US 6546507 to Coyle et al.**

28. Referring to claim 20, although Raimi in view of Grey does not specifically disclose the type and value information includes an incremental step size for each of the multiple parameters, using an incremental step for testing values is known in the art. An example of this is shown by Coyle, from line 46 of column 39, "The method 2850 starts at block 2852, where it verifies that the system operates correctly at a given initial value. The method 2850 in block 2854 tests whether the system passes at that value. If it does not, block 2856 reports a system failure. If the system passes, block 2858 adjusts the initial value to a new value, e.g., a single step up in the value. Block 2862 tests the system at this new value, and, if it passes, block 2862 saves the new value to a variable called HIGHGOOD. Then, method 2850 returns to block 2858 where the value can be again incremented in the same direction. If the test of block 2860 fails, the method 2850 proceeds to block 2864, where it resets the parameter to the initial value. Then, block 2866 tests whether the system passes at this value. If it does not, then block 2868 reports a system failure. If the system passes, block 2872 adjusts the parameter in the opposite direction to that of block 2858, e.g., a single step down. In other words, one of the blocks 2858 and 2872 increments the parameter value to test the operational limit in one direction, while the other decrements that value to test the operational limit in the other direction." A person of ordinary skill in the art at the time of the invention would have been motivated to incrementally step because as disclosed by Coyle, it permits operational envelope testing, and further allows a type of testing that further verifies system operation.

29. **Claim 46, 50 rejected under 35 U.S.C. 103(a) as being unpatentable over US 20040078674 to Raimi et al. in view of Official Notice.**

30. Referring to claim 46, Raimi discloses utilizing the state machine to perform system level validation of a processor (Field of invention.).

Although Raimi does not specifically disclose the processor is in "new silicon", implementing a processor in silicon is very well known in the art. Examiner takes official notice for silicon processors. A person of ordinary skill in the art at the time of the invention could have been motivated to have a silicon processor because silicon is a very common technique used to create processors.

31. Referring to claim 50, although Raimi does not specifically disclose pointers and "status functions" are utilized to build and maintain the state machine, pointers and status indicators are very well known in the art. Examiner takes official notice for pointers and status indicators. A person of ordinary skill in the art at the time of the invention could have been motivated to use pointers and status functions in building and maintaining a state machine because a state machine is a representation of functionality defined by states interrelated among other states. Pointers and status function thereby can, inter alia, indicate status and point to state.

Response to Arguments

32. Applicant's arguments with respect to the claims have been considered but are moot in view of the new ground(s) of rejection.

Conclusion

33. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure. See notice of references cited.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Gabriel L. Chu whose telephone number is (571) 272-3656. The examiner can normally be reached on weekdays between 8:30 AM and 5:00 PM.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Scott Baderman can be reached on (571) 272-3644. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

Gabriel L. Chu
Primary Examiner
Art Unit 2114

gc